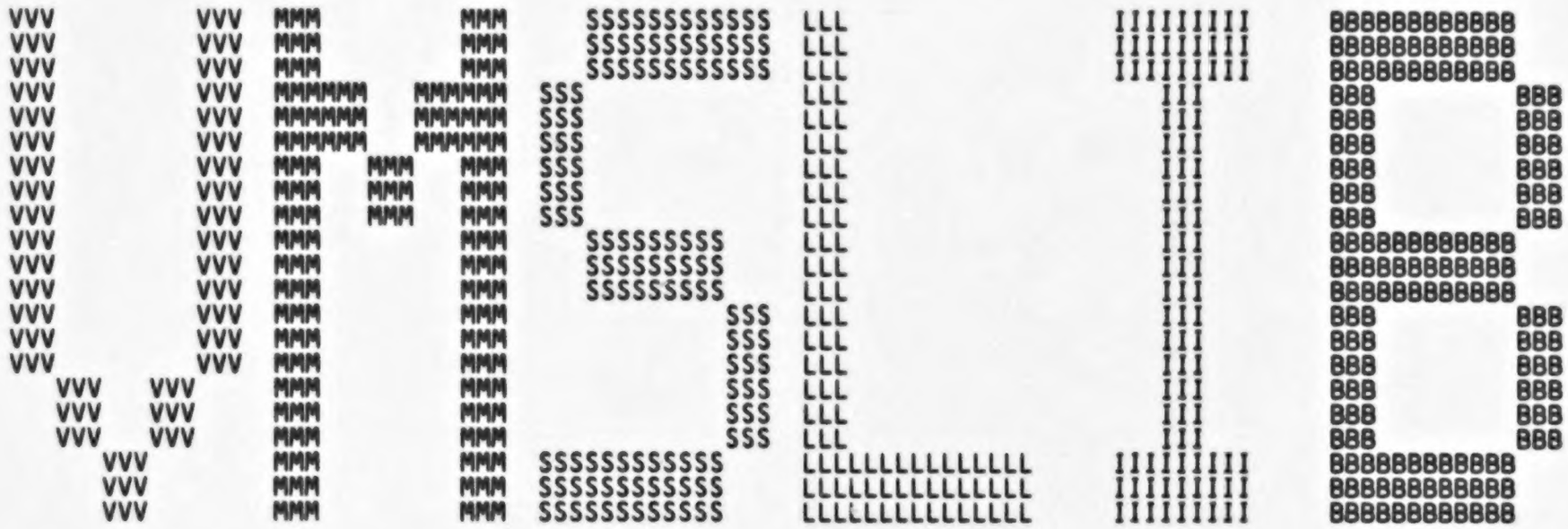
VVV VVV MMM MMM SSSSSSSSSSS LLL IIIIIIIII BBBBBBBBBBB
VVV VVV MMM MMM SSSSSSSSSSS LLL IIIIIIIII BBBBBBBBBBB
VVV VVV MMM MMM SSSSSSSSSSS LLL IIIIIIIII BBBBBBBBBBB
VVV VVV MMMMMM MMMMMM SSS LLL III BBB BBB
VVV VVV MMMMMM MMMMMM SSS LLL III BBB BBB
VVV VVV MMMMMM MMMMMM SSS LLL III BBB BBB
VVV VVV MMM MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSSSSSSSS LLL III BBBBBBBBBBB
VVV VVV MMM MMM SSSSSSSSS LLL III BBBBBBBBBBB
VVV VVV MMM MMM SSSSSSSSS LLL III BBBBBBBBBBB
VVV VVV MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSS LLL III BBB BBB
VVV MMM MMM SSSSSSSSSSS LLLLLLLLLLLLLLL IIIIIIIII BBBBBBBBBBB
VVV MMM MMM SSSSSSSSSSS LLLLLLLLLLLLLLL IIIIIIIII BBBBBBBBBBB
VVV MMM MMM SSSSSSSSSSS LLLLLLLLLLLLLLL IIIIIIIII BBBBBBBBBBB

FILEID**LIBCQMAT

```
LL            IIIIII    BBBBBBBB      CCCCCCCC    QQQQQQ    MM      MM    AAAAAA    TTTTTTTTTT
LL            IIIIII    BBBBBBBB      CCCCCCCC    QQQQQQ    MM      MM    AAAAAA    TTTTTTTTTT
LL              II      BB      BB    CC          QQ    QQ  MMMM  MMMM  AA    AA      TT
LL              II      BB      BB    CC          QQ    QQ  MMMM  MMMM  AA    AA      TT
LL              II      BB      BB    CC          QQ    QQ  MM  MM  MM  AA    AA      TT
LL              II      BB      BB    CC          QQ    QQ  MM  MM  MM  AA    AA      TT
LL              II      BBBBBBBB      CC          QQ    QQ  MM      MM  AA    AA      TT
LL              II      BBBBBBBB      CC          QQ    QQ  MM      MM  AA    AA      TT
LL              II      BB      BB    CC          QQ  QQ QQ  MM      MM  AAAAAAAAAA    TT
LL              II      BB      BB    CC          QQ  QQ QQ  MM      MM  AAAAAAAAAA    TT
LL              II      BB      BB    CC          QQ    QQ  MM      MM  AA    AA      TT          ....
LL              II      BB      BB    CC          QQ    QQ  MM      MM  AA    AA      TT          ....
LLLLLLLLLL    IIIIII    BBBBBBBB      CCCCCCCC    QQQQ  QQ  MM      MM  AA    AA      TT          ....
LLLLLLLLLL    IIIIII    BBBBBBBB      CCCCCCCC    QQQQ  QQ  MM      MM  AA    AA      TT          ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

Table of contents

```
0000     1          .TITLE  MATCH                   File specification matching
0000     2          .IDENT  'V04-000'
0000     3  ;---
0000     4  ;
0000     5  ;*****************************************************************************
0000     6  ;*                                                                           *
0000     7  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     8  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     9  ;*  ALL RIGHTS RESERVED.                                                     *
0000    10  ;*                                                                           *
0000    11  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
0000    12  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
0000    13  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
0000    14  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
0000    15  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
0000    16  ;*  TRANSFERRED.                                                             *
0000    17  ;*                                                                           *
0000    18  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
0000    19  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
0000    20  ;*  CORPORATION.                                                             *
0000    21  ;*                                                                           *
0000    22  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
0000    23  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    24  ;*                                                                           *
0000    25  ;*                                                                           *
0000    26  ;*****************************************************************************
0000    27
0000    28  ;++
0000    29  ; FACILITY:
0000    30  ;       Backup/Restore
0000    31  ;
0000    32  ; ABSTRACT:
0000    33  ;       This module contains file specification matching routines.
0000    34  ;
0000    35  ; ENVIRONMENT:
0000    36  ;       VAX/VMS user mode.
0000    37  ;
0000    38  ;
0000    39  ;--
0000    40  ;
0000    41  ; AUTHOR: M. Jack, CREATION DATE: 19-Sep-1980
0000    42  ; With acknowledgment to Goldstein and Halvorsen for the pieces.
0000    43  ; Chopped up by T. Krichevsky for inclusion in VMSLIB.
0000    44  ;
0000    45  ;
0000    46  ; MODIFIED BY:
0000    47  ;
0000    48  ;       V03-002 TSK0001         Tamar Krichevsky        15-Dec-1982
0000    49  ;               Modify PSECT attributes.
0000    50  ;
0000    51  ;       V03-001 MLJ0085         Martin L. Jack, 30-Mar-1982  13:40
0000    52  ;               Modify MATCH to avoid setting a wild directory bit past the
0000    53  ;               number of directory levels that actually exist in the
0000    54  ;               related file specification.
0000    55  ;
0000    56  ;       V02-007 MLJ0077         Martin L. Jack, 8-Feb-1982  15:11
0000    57  ;               Implement negative version numbers.
```

```
                     ;
                     ;       V02-006 MLJ0063         Martin L. Jack, 11-Dec-1981  14:41
                     ;               Support output UIC wildcarding by inserting a comma in the
                     ;               related filename string if the selection file specification
                     ;               is in UIC format.  Modify version number selection so that
                     ;               negative pattern versions are accepted and always match.
                     ;               Have INIT_NAMEBLOCK use expanded string if resultant is null.
                     ;
                     ;       V02-005 MLJ0054         Martin L. Jack, 6-Nov-1981  16:24
                     ;               Correct problem with leading [000000] in candidate directory.
                     ;               Modify calling sequence for MATCH_DIRECTORY to allow long
                     ;               filenames.
                     ;
                     ;       V02-004 MLJ0036         Martin L. Jack, 28-Aug-1981  16:58
                     ;               Extensive rewriting for reel restart and early scan termination.
                     ;
                     ;       V02-003 MLJ0025         Martin L. Jack, 8-May-1981  13:42
                     ;               Implement latest version selection.
                     ;
                     ;       V02-002 MLJ0022         Martin L. Jack, 21-Apr-1981  13:54
                     ;               Convert directory wildcarding to use RMS.
                     ;
                     ;       V02-001 MLJ0010         Martin L. Jack, 25-Mar-1981  14:04
                     ;               Add routines FAST_MATCH and PARSE_FAST_DIR.  Correct problems
                     ;               with [000000...] and the like as selection filespec.
                     ;
                     ;**
                     ;
                     ;
                     $NAMDEF
00000000             .PSECT  $CODE,RD,NOWRT,EXE,SHR,PIC
```

```
                     0000     90          .SBTTL  MATCH, match complete file specification
                     0000     91
                     0000     92 ;++
                     0000     93 ;
                     0000     94 ; Functional Description:
                     0000     95 ;       This routine executes a wild card match on a candidate and pattern
                     0000     96 ;       file specification.
                     0000     97 ;
                     0000     98 ; Calling Sequence:
                     0000     99 ;       CALLS/CALLG
                     0000    100 ;
                     0000    101 ; Input Parameters:
                     0000    102 ;       04(AP) = Address of descriptor for candidate file specification.
                     0000    103 ;       08(AP) = Address of descriptor for pattern file specification.
                     0000    104 ;
                     0000    105 ; Implicit Inputs:
                     0000    106 ;       none
                     0000    107 ;
                     0000    108 ; Output Parameters:
                     0000    109 ;       none
                     0000    110 ;
                     0000    111 ; Implicit Outputs:
                     0000    112 ;       none
                     0000    113 ;
                     0000    114 ; Routines Called:
                     0000    115 ;       MATCH_DIRECTORY
                     0000    116 ;       MATCH_FILENAME
                     0000    117 ;
                     0000    118 ; Routine Value:
                     0000    119 ;       True if the strings match.
                     0000    120 ;
                     0000    121 ; Signals:
                     0000    122 ;       none
                     0000    123 ;
                     0000    124 ; Side Effects:
                     0000    125 ;       none
                     0000    126 ;
                     0000    127 ;--
                     0000    128
                000C 0000    129          .ENTRY  MATCH,^M<R2,R3>
                     0002    130 ;
                     0002    131 ; Parse the candidate string into a directory string and a
                     0002    132 ; name, type, and version string.
                     0002    133 ;
     52  04 BC    7D 0002    134          MOVQ    @4(AP),R2                       ; Get candidate string descriptor
63   52  5B 8F    3A 0006    135          LOCC    #^A'[',R2,(R3)                  ; Scan for start of directory
             04    12 000B    136          BNEQ    10$                             ; Branch if found
   63    52  3C    3A 000D    137          LOCC    #^A'<',R2,(R3)                  ; Scan for alternate syntax
         52  70    9E 0011    138 10$:     MOVAB   -(R0),R2                        ; Prune beginning of string
      53  01 A1    9E 0014    139          MOVAB   1(R1),R3                        ;
63   52  5D 8F    3A 0018    140          LOCC    #^A']',R2,(R3)                  ; Scan for end of directory
             04    12 001D    141          BNEQ    20$                             ; Branch if found
   63    52  3E    3A 001F    142          LOCC    #^A'>',R2,(R3)                  ; Scan for alternate syntax
             53    DD 0023    143 20$:     PUSHL   R3                              ; Make directory descriptor
   7E    52  50    C3 0025    144          SUBL3   R0,R2,-(SP)                     ;
            01 A1    9F 0029    145          PUSHAB  1(R1)                           ; Make name, type, version descriptor
             70    9F 002C    146          PUSHAB  -(R0)                           ;
```

```
                                002E   147 ;
                                002E   148 ; Parse the pattern string into a directory string and a
                                002E   149 ; name, type, and version string.
                                002E   150 ;
      52    08 BC    7D        002E   151          MOVQ    @8(AP),R2                ; Get pattern string descriptor
63    52    5B 8F    3A        0032   152          LOCC    #^A'[',R2,(R3)           ; Scan for start of directory
                04    12        0037   153          BNEQ    30$                      ; Branch if found
   63    52    3C    3A        0039   154          LOCC    #^A'<',R2,(R3)           ; Scan for alternate syntax
         52    70    9E        003D   155 30$:     MOVAB   -(R0),R2                 ; Prune beginning of string
      53    01 A1    9E        0040   156          MOVAB   1(R1),R3                 ;
63    52    5D 8F    3A        0044   157          LOCC    #^A']',R2,(R3)           ; Scan for end of directory
                04    12        0049   158          BNEQ    40$                      ; Branch if found
   63    52    3E    3A        004B   159          LOCC    #^A'>',R2,(R3)           ; Scan for alternate syntax
                53    DD        004F   160 40$:     PUSHL   R3                       ; Make directory descriptor
   7E    52    50    C3        0051   161          SUBL3   R0,R2,-(SP)              ;
            01 A1    9F        0055   162          PUSHAB  1(R1)                    ; Make name, type, version descriptor
                70    9F        0058   163          PUSHAB  -(R0)                    ;
                                005A   164 ;
                                005A   165 ; Match the name, type, and version strings.
                                005A   166 ;
                5E    DD        005A   167          PUSHL   SP                       ; Push pattern descriptor
            14 AE    9F        005C   168          PUSHAB  20(SP)                   ; Push candidate descriptor
      72'AF    02    FB        005F   169          CALLS   #2,B^MATCH_FILENAME      ; Match filenames
            0B 50    E9        0063   170          BLBC    R0,50$                   ; Branch if failed
                                0066   171 ;
                                0066   172 ; Match the directory strings.
                                0066   173 ;
            08 AE    9F        0066   174          PUSHAB  8(SP)                    ; Push pattern descriptor
            1C AE    9F        0069   175          PUSHAB  28(SP)                   ; Push candidate descriptor
   00E3'CF    02    FB        006C   176          CALLS   #2,W^MATCH_DIRECTORY     ; Match directories
                04    0071   177 50$:     RET                              ; Return
```

```
                     0072   179          .SBTTL  MATCH_FILENAME, match file name, type, version
                     0072   180
                     0072   181 ;++
                     0072   182 ;
                     0072   183 ; Functional Description:
                     0072   184 ;       This routine executes a wild card match on the name, type, and version
                     0072   185 ;       of a candidate and pattern file specification.  If the pattern selects
                     0072   186 ;       latest version, this routine assumes that the candidate file is the
                     0072   187 ;       latest version, and therefore that the version matches.
                     0072   188 ;
                     0072   189 ; Calling Sequence:
                     0072   190 ;       CALLS/CALLG
                     0072   191 ;
                     0072   192 ; Input Parameters:
                     0072   193 ;       04(AP) = Address of descriptor for candidate file name, type, version.
                     0072   194 ;       08(AP) = Address of descriptor for pattern file name, type, version.
                     0072   195 ;
                     0072   196 ; Implicit Inputs:
                     0072   197 ;       none
                     0072   198 ;
                     0072   199 ; Output Parameters:
                     0072   200 ;       none
                     0072   201 ;
                     0072   202 ; Implicit Outputs:
                     0072   203 ;       none
                     0072   204 ;
                     0072   205 ; Routines Called:
                     0072   206 ;       FMG$MATCH_NAME
                     0072   207 ;       LIB$CVT_DTB
                     0072   208 ;
                     0072   209 ; Routine Value:
                     0072   210 ;       True if the strings match.
                     0072   211 ;
                     0072   212 ; Signals:
                     0072   213 ;       none
                     0072   214 ;
                     0072   215 ; Side Effects:
                     0072   216 ;       none
                     0072   217 ;
                     0072   218 ;--
                     0072   219
             03FC    0072   220          .ENTRY  MATCH_FILENAME,^M<R2,R3,R4,R5,R6,R7,R8,R9>
                     0074   221 ;
                     0074   222 ; Parse the candidate string into a name and type string and a version string.
                     0074   223 ;
   52  04 BC   7D    0074   224          MOVQ    @4(AP),R2                       ; Get candidate string descriptor
      52   52   3C    0078   225          MOVZWL  R2,R2                           ; Truncate length to a word
63    52   3B   3A    007B   226          LOCC    #^A':',R2,(R3)                  ; Scan for version number
        61   13   13    007F   227          BEQL    20$                             ; Branch if not found
      52   50   C2    0081   228          SUBL2   R0,R2                           ; R2-R3: candidate name and type
57    01 A1   9E    0084   229          MOVAB   1(R1),R7                        ; R6-R7: candidate version
      56   70   9E    0088   230          MOVAB   -(R0),R6                        ;
                     008B   231 ;
                     008B   232 ; Parse the pattern string into a name and type string and a version string.
                     008B   233 ;
54  08 BC   7D    008B   234          MOVQ    @8(AP),R4                       ; Get pattern string descriptor
      54   54   3C    008F   235          MOVZWL  R4,R4                           ; Truncate length to a word
```

```
                                00A2    241 ;
                                00A2    242 ; Match the file name and type.
                                00A2    243 ;
00000000'GF   16                00A2    244         JSB     G^FMG$MATCH_NAME        ; Match name and type
      37 50   E9                00A8    245         BLBC    R0,20$                  ; Branch if failed
                                00AB    246 ;
                                00AB    247 ; The match succeeded.  Now the version numbers must be matched.
                                00AB    248 ;
              58    D5            00AB    249         TSTL    R8                      ; Pattern version null?
              30    13            00AD    250         BEQL    10$                     ; Branch if yes
           2A     69    91        00AF    251         CMPB    (R9),#^A'*'             ; Wildcard pattern version?
              2B    13            00B2    252         BEQL    10$                     ; Branch if yes
              7E    DF            00B4    253         PUSHAL  -(SP)                   ; Push address of result location
           7E     58    7D        00B6    254         MOVQ    R8,-(SP)                ; Push descriptor
00000000'GF       03    FB        00B9    255         CALLS   #3,G^LIB$CVT_DTB        ; Convert pattern version number
         1F 50    E9              00C0    256         BLBC    R0,20$                  ; Branch if conversion failed
              58    8E    D0      00C3    257         MOVL    (SP)+,R8                ; Get pattern version number
              17    15            00C6    258         BLEQ    10$                     ; Branch if explicit zero or negative
              7E    DF            00C8    259         PUSHAL  -(SP)                   ; Push address of result location
           7E     56    7D        00CA    260         MOVQ    R6,-(SP)                ; Push descriptor
00000000'GF       03    FB        00CD    261         CALLS   #3,G^LIB$CVT_DTB        ; Convert candidate version number
         0B 50    E9              00D4    262         BLBC    R0,20$                  ; Branch if conversion failed
              50    D4            00D7    263         CLRL    R0                      ; Assume failure return
              58    8E    D1      00D9    264         CMPL    (SP)+,R8                ; Compare versions
              01    13            00DC    265         BEQL    10$                     ; Branch if versions equal
                    04            00DE    266         RET                             ; Return failure
50 01         D0                  00DF    267 10$:    MOVL    #1,R0                   ; Set success status
                    04            00E2    268 20$:    RET                             ; Return
```

```
                     00E3   270          .SBTTL  MATCH_DIRECTORY, match directories
                     00E3   271
                     00E3   272 ;++
                     00E3   273 ;
                     00E3   274 ; Functional Description:
                     00E3   275 ;       This routine executes a wild card match on the directory
                     00E3   276 ;       of a candidate and pattern file specification.
                     00E3   277 ;
                     00E3   278 ; Calling Sequence:
                     00E3   279 ;       CALLS/CALLG
                     00E3   280 ;
                     00E3   281 ; Input Parameters:
                     00E3   282 ;       04(AP) = Address of descriptor for candidate directory.
                     00E3   283 ;       08(AP) = Address of descriptor for pattern directory.
                     00E3   284 ;       20(AP) = Optional address of descriptor for pattern
                     00E3   285 ;                name, type, and version.
                     00E3   286 ;
                     00E3   287 ; Implicit Inputs:
                     00E3   288 ;       none
                     00E3   289 ;
                     00E3   290 ; Output Parameters:
                     00E3   291 ;       12(AP) = Optional address of descriptor to receive terminator file
                     00E3   292 ;                name and type.
                     00E3   293 ;       16(AP) = Optional address of word to receive terminator file version
                     00E3   294 ;                number.
                     00E3   295 ;
                     00E3   296 ; Implicit Outputs:
                     00E3   297 ;       none
                     00E3   298 ;
                     00E3   299 ; Routines Called:
                     00E3   300 ;       FMG$MATCH_NAME
                     00E3   301 ;       LIB$CVT_DTB
                     00E3   302 ;       PARSE_DIRECTORY
                     00E3   303 ;
                     00E3   304 ; Routine Value:
                     00E3   305 ;       Bit 0:  Set if the directory strings match.
                     00E3   306 ;       Bit 1:  Set if the directory must be scanned.
                     00E3   307 ;
                     00E3   308 ; Signals:
                     00E3   309 ;       none
                     00E3   310 ;
                     00E3   311 ; Side Effects:
                     00E3   312 ;       none
                     00E3   313 ;
                     00E3   314 ;--
                     00E3   315
                03FC 00E3   316          .ENTRY  MATCH_DIRECTORY,^M<R2,R3,R4,R5,R6,R7,R8,R9>
5E   FF38 CE      9E 00E5   317          MOVAB   -200(SP),SP             ; Make directory parse space on stack
                     00EA   318 ;
                     00EA   319 ; Parse the candidate directory string.
                     00EA   320 ;
52     04 BC      7D 00EA   321          MOVQ    @4(AP),R2               ; Get candidate string descriptor
   52    52      3C 00EE   322          MOVZWL  R2,R2                   ; Truncate length to a word
   54    5E      D0 00F1   323          MOVL    SP,R4                   ; Point to result area
      0123      30 00F4   324          BSBW    PARSE_DIRECTORY         ; Parse the specification
                     00F7   325 ;
                     00F7   326 ; Parse the pattern directory string.
```

```
                     00F7   327 ;
 52   08 BC    7D     00F7   328          MOVQ    @8(AP),R2               ; Get pattern string descriptor
      52   52    3C     00FB   329          MOVZWL  R2,R2                   ; Truncate length to a word
 54   44 AE    9E     00FE   330          MOVAB   68(SP),R4               ; Point to result area
        0115    30     0102   331          BSBW    PARSE_DIRECTORY         ; Parse the specification
                     0105   332 ;
                     0105   333 ; Set up for matching directories.
                     0105   334 ;
      52   6E    3C     0105   335          MOVZWL  (SP),R2                 ; Get candidate descriptor count
 53   04 AE    9E     0108   336          MOVAB   4(SP),R3                ; Point to candidate string results
 54   44 AE    3C     010C   337          MOVZWL  68(SP),R4               ; Get pattern descriptor count
 55   48 AE    9E     0110   338          MOVAB   72(SP),R5               ; Point to first pattern descriptor
           56    7C     0114   339          CLRQ    R6                      ; Clear saved directory count, pointer
                     0116   340 ;
                     0116   341 ; A special matching rule applies if the pattern string directory is in
                     0116   342 ; UIC format.  Such a directory matches only if the target string
                     0116   343 ; directory contains six octal digits.
                     0116   344 ;
    19 46 AE    E9     0116   345          BLBC    70(SP),20$              ; Branch if pattern not UIC format
     01   52    D1     011A   346          CMPL    R2,#1                   ; Exactly one directory in target?
          45    12     011D   347          BNEQ    50$                     ; Branch if no
     50   63    7D     011F   348          MOVQ    (R3),R0                 ; Get target directory descriptor
     06   50    D1     0122   349          CMPL    R0,#6                   ; Six characters?
          3D    12     0125   350          BNEQ    50$                     ; Branch if no
59   81   30    83     0127   351 10$:     SUBB3   #^A'0',(R1)+,R9         ; Bias character by ASCII 0
     07   59    91     012B   352          CMPB    R9,#7                   ; Make sure in range
          34    1A     012E   353          BGTRU   50$                     ; Branch if no
        F4 50    F5     0130   354          SOBGTR  R0,10$                  ; Loop until all examined
                     0133   355 ;
                     0133   356 ; Handle MFD in pattern and candidate strings.
                     0133   357 ;
     06   65    D1     0133   358 20$:     CMPL    (R5),#6                 ; First pattern directory length 6?
          1A    12     0136   359          BNEQ    40$                     ; Branch if no
04 B5  06   30    3B     0138   360          SKPC    #^A'0',#6,@4(R5)        ; First pattern directory '000000'?
          13    12     013D   361          BNEQ    40$                     ; Branch if no
     06   63    D1     013F   362          CMPL    (R3),#6                 ; First candidate directory length 6?
          07    12     0142   363          BNEQ    30$                     ; Branch if no
04 B3  06   30    3B     0144   364          SKPC    #^A'0',#6,@4(R3)        ; First candidate directory '000000'?
          1C    13     0149   365          BEQL    60$                     ; Branch if yes
          54    D7     014B   366 30$:     DECL    R4                      ; Prune '000000' from pattern
     55   08    C0     014D   367          ADDL2   #8,R5                   ;
          15    11     0150   368          BRB     60$                     ; Go to do full match
     06   63    D1     0152   369 40$:     CMPL    (R3),#6                 ; First candidate directory length 6?
          10    12     0155   370          BNEQ    60$                     ; Branch if no
04 B3  06   30    3B     0157   371          SKPC    #^A'0',#6,@4(R3)        ; Candidate directory is '000000'?
          09    12     015C   372          BNEQ    60$                     ; Branch if no
     50   65    7D     015E   373          MOVQ    (R5),R0                 ; Get descriptor for first pattern
        008F    31     0161   374          BRW     140$                    ; Branch to fail with scan
        0089    31     0164   375 50$:     BRW     130$                    ; Branch to fail
                     0167   376 ;
                     0167   377 ; Now execute the full-scale match.
                     0167   378 ;
          54    D7     0167   379 60$:     DECL    R4                      ; Pattern exhausted?
          31    19     0169   380          BLSS    80$                     ; Branch if yes
     50   85    7D     016B   381          MOVQ    (R5)+,R0                ; Get next directory in pattern
     2E   61    91     016E   382          CMPB    (R1),#^A'.'             ; Check for ellipsis
          6C    13     0171   383          BEQL    110$                    ; Branch if yes
```

```
                52   D7  0173   384          DECL    R2                              ; Candidate exhausted?
                7C   19  0175   385          BLSS    140$                            ; Branch if yes
                3C   BB  0177   386          PUSHR   #^M<R2,R3,R4,R5>                ; Save registers around MATCH_NAME
          54    50   7D  0179   387          MOVQ    R0,R4                           ; Load pattern descriptor to R4-R5
          52    63   7D  017C   388          MOVQ    (R3),R2                         ; Load candidate descriptor to R2-R3
00000000'GF    16   017F   389          JSB     G^FMG$MATCH_NAME                ; Check candidate against pattern
                3C   BA  0185   390          POPR    #^M<R2,R3,R4,R5>                ; Restore registers
          53    08   C0  0187   391          ADDL2   #8,R3                           ; Advance past descriptor
             DA 50   E8  018A   392          BLBS    R0,60$                          ; Branch if match
                         018D   393 ;
                         018D   394 ; We have detected a mismatch, or we are out of pattern string while there
                         018D   395 ; is input left. Back up to the last ellipsis, advance a directory of the
                         018D   396 ; input, and try again.
                         018D   397 ;
                56   D7  018D   398 70$:     DECL    R6                              ; Count a directory from saved input
                5B   19  018F   399          BLSS    120$                            ; Branch if no saved input
          57    08   C0  0191   400          ADDL2   #8,R7                           ; Set to try next input directory
          52    56   7D  0194   401          MOVQ    R6,R2                           ; Restore pointers to backup point
          54    58   7D  0197   402          MOVQ    R8,R4                           ;     to retry matching
                CB   11  019A   403          BRB     60$                             ; Continue testing
                         019C   404 ;
                         019C   405 ; Here when pattern string is exhausted.
                         019C   406 ;
                52   D5  019C   407 80$:     TSTL    R2                              ; Input exhausted?
                ED   12  019E   408          BNEQ    70$                             ; Branch if no
                         01A0   409 ;
                         01A0   410 ; Establish the pattern file name, type, and version specified by
                         01A0   411 ; the fourth parameter as the terminator and return success.
                         01A0   412 ;
          05    6C   91  01A0   413 90$:     CMPB    (AP),#5                         ; Parameters present?
                36   1F  01A3   414          BLSSU   100$                            ; Branch if no
       59   0C   AC   D0  01A5   415          MOVL    12(AP),R9                       ; Point to result area descriptor
                69   B4  01A9   416          CLRW    (R9)                            ; Set no terminator
                57   D5  01AB   417          TSTL    R7                              ; Ellipsis found in string?
                2C   12  01AD   418          BNEQ    100$                            ; Branch if yes
       52   14   BC   7D  01AF   419          MOVQ    @20(AP),R2                      ; Get descriptor for specification
          52    52   3C  01B3   420          MOVZWL  R2,R2                           ; Truncate length to a word
    63    52    3B   3A  01B6   421          LOCC    #^A':',R2,(R3)                  ; Scan for version number
                1F   13  01BA   422          BEQL    100$                            ; Branch if not found
          52    50   C2  01BC   423          SUBL2   R0,R2                           ; R2-R3: name and type
                00   DD  01BF   424          PUSHL   #0                              ; Create and clear result location
                5E   DD  01C1   425          PUSHL   SP                              ; Push address of result location
             01 A1   9F  01C3   426          PUSHAB  1(R1)                           ; Push address of version number
                70   9F  01C6   427          PUSHAB  -(R0)                           ; Push length of version number
00000000'GF     03   FB  01C8   428          CALLS   #3,G^LIB$CVT_DTB                ; Convert version number
       10 BC    8E   F7  01CF   429          CVTLW   (SP)+,@16(AP)                   ; Set terminator file version
          69    52   B0  01D3   430          MOVW    R2,(R9)                         ; Set byte count of name and type
 04 B9    63    52   28  01D6   431          MOVC3   R2,(R3),@4(R9)                  ; Set terminator file name and type
          50    03   D0  01DB   432 100$:    MOVL    #3,R0                           ; Set success status
                04   01DE   433          RET                                     ; Return
                         01DF   434 ;
                         01DF   435 ; We have encountered an ellipsis in the pattern string.  Save the string
                         01DF   436 ; pointers for backup and retry.
                         01DF   437 ;
          56    52   7D  01DF   438 110$:    MOVQ    R2,R6                           ; Save current string pointers
          58    54   7D  01E2   439          MOVQ    R4,R8                           ;     of both strings
                54   D5  01E5   440          TSTL    R4                              ; Pattern string null after ellipsis?
```

```
                B7   13   01E7    441          BEQL    90$                     ; Ellipsis at end matches all
              FF7B   31   01E9    442          BRW     60$                     ; Continue testing
                     01EC    443 ;
                     01EC    444 ; Here when match fails.
                     01EC    445 ;
                57   D5   01EC    446 120$:    TSTL    R7                      ; Ellipsis found in string?
                03   12   01EE    447          BNEQ    140$                    ; Branch if yes
                50   D4   01F0    448 130$:    CLRL    R0                      ; Set failure
                     04   01F2    449          RET
                     01F3    450 ;
                     01F3    451 ; Here to establish the directory specified by R0-R1 as the terminator.
                     01F3    452 ;
          04    6C   91   01F3    453 140$:    CMPB    (AP),#4                 ; Parameters present?
                1E   1F   01F6    454          BLSSU   150$                    ; Branch if no
       59    0C AC   D0   01F8    455          MOVL    12(AP),R9               ; Point to result area descriptor
                69   B4   01FC    456          CLRW    (R9)                    ; Set no terminator
                57   D5   01FE    457          TSTL    R7                      ; Ellipsis found in string?
                14   12   0200    458          BNEQ    150$                    ; Branch if yes
    69    50    04   A1   0202    459          ADDW3   #4,R0,(R9)              ; Set byte count of name and type
 04 B9    61    50   28   0206    460          MOVC3   R0,(R1),@4(R9)          ; Set terminator file name
63   5249442E   8F   D0   020B    461          MOVL    #^A'.DIR',(R3)          ; Set terminator file type
      10 BC     01   B0   0212    462          MOVW    #1,@16(AP)              ; Set terminator file version
          50    02   D0   0216    463 150$:    MOVL    #2,R0                   ; Set no match, scan directory
                     04   0219    464          RET                             ; Return
```

```
                     021A   466          .SBTTL  PARSE_DIRECTORY, parse directory into components
                     021A   467
                     021A   468 ;++
                     021A   469 ;
                     021A   470 ; Functional Description:
                     021A   471 ;       This routine parses the directory portion of a file specification.
                     021A   472 ;
                     021A   473 ; Calling Sequence:
                     021A   474 ;       JSB
                     021A   475 ;
                     021A   476 ; Input Parameters:
                     021A   477 ;       R2 = Length of directory string.
                     021A   478 ;       R3 = Address of directory string.
                     021A   479 ;       R4 = Pointer to result area.
                     021A   480 ;
                     021A   481 ; Implicit Inputs:
                     021A   482 ;       none
                     021A   483 ;
                     021A   484 ; Output Parameters:
                     021A   485 ;       none
                     021A   486 ;
                     021A   487 ; Implicit Outputs:
                     021A   488 ;       Result area contains a count of descriptors followed by one descriptor
                     021A   489 ;       for each component of the directory specification.  An ellipsis is
                     021A   490 ;       represented by a one-byte string '.'.  If the directory is in UIC
                     021A   491 ;       format, bit 16 of the descriptor count longword is set.
                     021A   492 ;
                     021A   493 ; Routines Called:
                     021A   494 ;       none
                     021A   495 ;
                     021A   496 ; Routine Value:
                     021A   497 ;       none
                     021A   498 ;
                     021A   499 ; Signals:
                     021A   500 ;       none
                     021A   501 ;
                     021A   502 ; Side Effects:
                     021A   503 ;       R0-R5 destroyed.
                     021A   504 ;
                     021A   505 ;--
                     021A   506
                     021A   507 PARSE_DIRECTORY:
          55   84    DE   021A   508          MOVAL   (R4)+,R5                ; Keep pointer to result area and bump
               65    D4   021D   509          CLRL    (R5)                    ; Clear component count
                     021F   510 ;
                     021F   511 ; Main scanning loop.
                     021F   512 ;
    63    52   2E    3A   021F   513 10$:     LOCC    #^A'.',R2,(R3)          ; Scan for delimiter
               29    13   0223   514          BEQL    20$                     ; Branch if none found
    84    52   50    C3   0225   515          SUBL3   R0,R2,(R4)+             ; Set length of this component
          84   53    D0   0229   516          MOVL    R3,(R4)+                ; Set address of this component
               65    D6   022C   517          INCL    (R5)                    ; Count this component
               52    70   9E   022E   518          MOVAB   -(R0),R2                ; Prune this component from string
      53    01   A1   9E   0231   519          MOVAB   1(R1),R3                ;
               02    52   D1   0235   520          CMPL    R2,#2                   ; At least 2 characters left?
                     E5   19   0238   521          BLSS    10$                     ; Branch if no
  2E2E 8F    63   B1   023A   522          CMPW    (R3),#^A'..'            ; Ellipsis present?
```

```
                DE   12   023F   523          BNEQ    10$                     ; Branch if no
             52 02   C2   0241   524          SUBL2   #2,R2                   ; Adjust count
             B4 01   D0   0244   525          MOVL    #1,(R4)+                ; Set length of this component
             84 83   3E   0247   526          MOVAW   (R3)+,(R4)+             ; Set address of '.'
                65   D6   024A   527          INCL    (R5)                    ; Count this component
                D1   11   024C   528          BRB     10$                     ; Branch to get next component
                          024E   529 ;
                          024E   530 ; Here when no '.' found in string.
                          024E   531 ;
                52   D5   024E   532 20$:     TSTL    R2                      ; At end of string?
                05   13   0250   533          BEQL    30$                     ; Branch if no
             84 52   7D   0252   534          MOVQ    R2,(R4)+                ; Set descriptor for last component
                65   D6   0255   535          INCL    (R5)                    ; Count last component
             01 65   D1   0257   536 30$:     CMPL    (R5),#1                 ; One component?
                5D   12   025A   537          BNEQ    70$                     ; Branch if no
          52 04 A5   7D   025C   538          MOVQ    4(R5),R2                ; Get descriptor for first component
       63 52   2C   3A   0260   539          LOCC    #^A',',R2,(R3)          ; In UIC format?
                53   13   0264   540          BEQL    70$                     ; Branch if no
                          0266   541 ;
                          0266   542 ; Special processing for UIC-format directory.
                          0266   543 ;
          02 A5   01   88   0266   544          BISB2   #1,2(R5)                ; Set UIC format bit
          54 0C A5   9E   026A   545          MOVAB   12(R5),R4               ; Point to UIC storage
          04 A5   06   D0   026E   546          MOVL    #6,4(R5)                ; Reset component descriptor
          08 A5   54   D0   0272   547          MOVL    R4,8(R5)                ;
    B4 30303030   8F   D0   0276   548          MOVL    #^A'0000',(R4)+         ; Initialize to 000000
       84   3030   8F   B0   027D   549          MOVW    #^A'00',(R4)+           ;
                53   52   C0   0282   550          ADDL2   R2,R3                   ; Point past member number
                52   50   C2   0285   551          SUBL2   R0,R2                   ; Get length of group number
                50   D7   0288   552          DECL    R0                      ; Get length of member number
          2A FF   A3   91   028A   553          CMPB    -1(R3),#^A'*'           ; Is member number '*'?
                0A   12   028E   554          BNEQ    40$                     ; Branch if no
    74 2525   8F   B0   0290   555          MOVW    #^A'%%',-(R4)           ; Make it '%%%' in output
          74   25   90   0295   556          MOVB    #^A'%',-(R4)            ;
                06   11   0298   557          BRB     50$                     ; Done
          74 73   90   029A   558 40$:     MOVB    -(R3),-(R4)             ; Copy one digit
             FA 50   F5   029D   559          SOBGTR  R0,40$                  ; Loop until done
       54 0F A5   9E   02A0   560 50$:     MOVAB   15(R5),R4               ; Point past group number
       2A FF A1   91   02A4   561          CMPB    -1(R1),#^A'*'           ; Is group number '*'?
                09   12   02A8   562          BNEQ    60$                     ; Branch if no
    74 2525   8F   B0   02AA   563          MOVW    #^A'%%',-(R4)           ; Make it '%%%' in output
          74   25   90   02AF   564          MOVB    #^A'%',-(R4)            ;
                05   02B2   565          RSB                             ; Done
          74 71   90   02B3   566 60$:     MOVB    -(R1),-(R4)             ; Copy one digit
             FA 52   F5   02B6   567          SOBGTR  R2,60$                  ; Loop till done
                05   02B9   568 70$:     RSB                             ; Done
                          02BA   569
                          02BA   570          .END
```

```
FMG$MATCH_NAME                  ********    X   02
LIB$CVT_DTB                     ********    X   02
MATCH                           00000000 RG     02
MATCH_DIRECTORY                 000000E3 RG     02
MATCH_FILENAME                  00000072 RG     02
PARSE_DIRECTORY                 0000021A R      02
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
ABS	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
$CODE	000002BA (698.)	02 (2.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	40	00:00:00.08	00:00:00.50
Command processing	137	00:00:00.49	00:00:02.73
Pass 1	152	00:00:02.84	00:00:07.97
Symbol table sort	0	00:00:00.20	00:00:00.40
Pass 2	129	00:00:01.33	00:00:04.71
Symbol table output	4	00:00:00.02	00:00:00.05
Psect synopsis output	5	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	469	00:00:04.98	00:00:16.38

```
The working set limit was 900 pages.
17435 bytes (35 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 164 non-local and 29 local symbols.
570 source lines were read in Pass 1, producing 20 object records in Pass 2.
8 pages of virtual memory were used to define 7 macros.
```

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYSLIB]STARLET.MLB;2	4

```
217 GETS were required to define 4 macros.

There were no errors, warnings or information messages.

MACRO/DISA=TRACE/LIS=LIS$:LIBCQMAT/OBJ=OBJ$:LIBCQMAT MSRC$:LIBCQMAT/UPDATE=(ENH$:LIBCQMAT)
```

0435 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

LIBACP
LIS

CALLIMAGE
LIS

DISMOUMSG
LIS

LIBCLICAL
LIS

LIBCOMAT
LIS

FILEDFLT
LIS

LIBASSIGN
LIS

CORMSG
LIS

CVTATB
LIS

LIBCQUAL
LIS

INITMSG
LIS

CREMAPSEC
LIS